

IMPROVING FRACTAL IMAGE COMPRESSION SCHEMES THROUGH QUANTIZATION AND ENTROPY CODING

M. Ghazel¹, A.K. Khandani¹ and E.R. Vrscay²

¹Department of Electrical and Computer Engineering

²Department of Applied Mathematics

University of Waterloo

Waterloo, ON, Canada N2L 3G1

ghazel@shannon.uwaterloo.ca khandani@shannon.uwaterloo.ca

ervrscay@links.uwaterloo.ca <http://links.uwaterloo.ca>

ABSTRACT

We explore the transform coefficients of various fractal-based schemes for statistical dependence and exploit correlations to improve the compression capabilities of these schemes. In most of the standard fractal-based schemes, the transform coefficients exhibit a degree of linear dependence that can be exploited by using an appropriate vector quantizer such as the LBG algorithm. Additional compression is achieved by lossless Huffman coding of the quantized coefficients.

1. INTRODUCTION

JPEG and Vector Quantization have certainly been the most commonly used methods for image compression, followed more recently by wavelet methods. However, over the past few years, there has also been much interest and development in fractal-based methods of image compression [3, 4, 7], accompanied by a number of significant developments [5]. Most of these schemes are centered around the method of Iterated Function Systems [2]. The common conception is that fractal-based schemes exploit geometric self-similarities that are inherent in images. However, a more realistic justification of these schemes is that they exploit the local scaling properties of irregularities, e.g. edges, across a range of resolutions, in turn reflected in the decay of associated wavelet coefficients.[8, 12]

Fractal-based schemes seek to approximate a target image function $f(x, y)$ as a union or *tiling* of geometrically shrunken copies of subsets of f with suitably transformed grey level values. As such, the image f is approximated by the fixed point \bar{f} of a contractive operator or *fractal transform* T . From Banach's Fixed Point Theorem, the approximation \bar{f} may be generated by the iteration procedure $f_{n+1} = Tf_n$. (f_0 can

be a blank screen, for example.) It is the transform T , in particular the coefficients of its geometric and grey-level transformations, that is stored in computer memory, often requiring much less storage than the original image f , resulting in significant (lossy) data compression.

In this study, we explore the statistical dependencies of transform coefficients for various fractal schemes and exploit such correlations to improve their compression capabilities. We find that the fractal transform coefficients for most standard fractal-based schemes exhibit a degree of linear dependence that can be exploited by using a proper vector quantizer such as the LBG algorithm. Additional compression is achieved by lossless Huffman coding of the quantized coefficients.

2. BASICS OF FRACTAL COMPRESSION

As stated earlier, given a target image f , we seek to find a contractive fractal transform operator T with fixed point \bar{f} approximates f to some suitable degree of accuracy, i.e. $d(\bar{f}, f) < \epsilon$ where d denotes \mathcal{L}^2 (RMS) metric. From an important corollary of Banach's Fixed Point Theorem, referred to in the IFS literature as the Collage Theorem [1], this inverse problem reduces to the following: Find an operator T such that $d(f, T(f))$ is suitably small. In other words, T should send the target close to itself.

Originally, IFS-type methods sought to express a target set or image as a union of shrunken copies of itself. However, most real-world objects are rarely so entirely self-similar. Instead, self-similarity may be exhibited only locally, in the sense that subregions of an image may be self-similar. This is the basis of the *block-encoding scheme* of Jacquin [6] that has been used in most fractal-based methods to date and which we briefly summarize below:

- The target image is subdivided into two different partitions of nonoverlapping blocks, as illustrated in figure 1.

1. $M \times M$ parent blocks, P_{ij} , $i, j = 1, \dots, M$, for simplicity we typically choose $M = 2^m$, for some integer m .

2. $N \times N$ child blocks, C_{kl} , $k, l = 1, \dots, N$, typically, $N = 2 \times M$.

- For each child block, C_{kl} , $k, l = 1, \dots, N$, choose a parent block P_{ij} and one of eight possible geometric contractions $w_{ijkl}^{(m)} : C_{ij} \rightarrow P_{kl}$, $1 \leq m \leq 8$.

- Find a *grey level map* $\phi_{ijkl}^{(m)} : \mathbf{R} \rightarrow \mathbf{R}$ that maps the grey level values on P_{ij} onto the grey level values on C_{kl} in an optimal way, i.e. by minimizing the \mathcal{L}^2 distance

$$\|f(w_{ijkl}^{(m)}(x, y)) - \phi_{ijkl}^{(m)}(f(x, y))\|_2. \quad (1)$$

In standard practice, the grey level maps are assumed to be affine, i.e.

$$\phi(t) = \alpha t + \beta.$$

Because the child blocks are nonoverlapping the minimization of the \mathcal{L}^2 distance in Eq. (1) becomes a least squares determination of α and β .

- For each child block C_{kl} , the parent block indices $(i(k, l), j(k, l))$, the geometric map index $m \in \{1, 2, \dots, 8\}$ and corresponding grey level coefficients $(\alpha_{ij}, \beta_{ij})$ define the *fractal code* for C_{kl} .

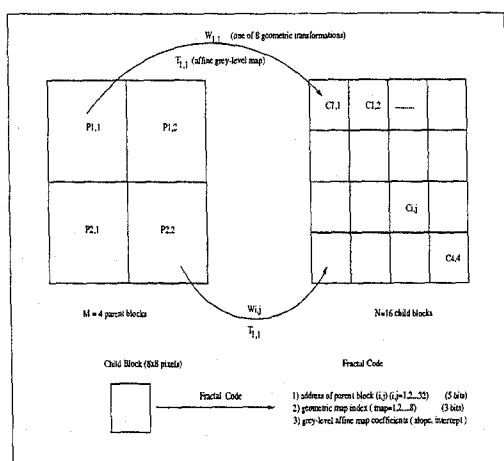


Figure 1: Fractal block-coding of an image: a parent block P_{ij} is mapped onto a child block C_{kl} .

Ideally, the best results are obtained from an *exhaustive search* in which *all* parent blocks as well as

all possible geometric maps are tested: For each child C_{kl} , the parent block address, the geometric map index and grey level map coefficients yielding the smallest \mathcal{L}^2 distance in Eq.(1) determine the fractal code for C_{kl} . This procedure, introduced by Jacquin [6], is obviously a computationally expensive procedure. Restrictive searches requiring much less computational time often yield results with rather small sacrifices in accuracy. This has been the object of much research in fractal image compression [5]. Some standard approaches include: (a) a reduced search of parent blocks to a prescribed neighborhood of the child block, (b) consideration of only a few - perhaps only one - geometric transformations of the parent block to the child block, (c) consideration only of parent blocks that overlap a child block, but allowing for “sliding”. (d) modified grey level maps, including the place-dependent maps used in the so-called Bath Fractal Transform (BFT) [9, 10].

In what follows, we consider three simple variations of the Jacquin block-encoding scheme along with the BFT, as summarized in Table 1.

Scheme	Parent Search?	All geometric maps?
I	Exhaustive	Yes
II	Exhaustive	No
III	No	No
BFT	No	No

Table 1: Fractal schemes studied in this paper

In the case of no parent searching, a child block is simply matched with its “co-centric” parent block, i.e. the parent block centered at the same pixel as the child block, except in the case of border blocks. The place-dependent BFT grey level maps used in this study have the form

$$\phi(x, y, f(x, y)) = \alpha f(x, y) + \beta + \gamma_x x + \gamma_y y. \quad (2)$$

The final three terms correspond to the addition of a planar or “ramping” term to the fractal term $\alpha f(x, y)$.

3. QUANTIZATION OF FRACTAL COEFFICIENTS

In order to improve compression, we explored the fractal coefficients for various types of statistical dependence. It has been found that for the first three schemes, the grey-level coefficients exhibit a high degree of linear dependence as illustrated in figure 2. Similar correlation was found to exist among some of the BFT coefficients. This suggests the use of an appropriate vector

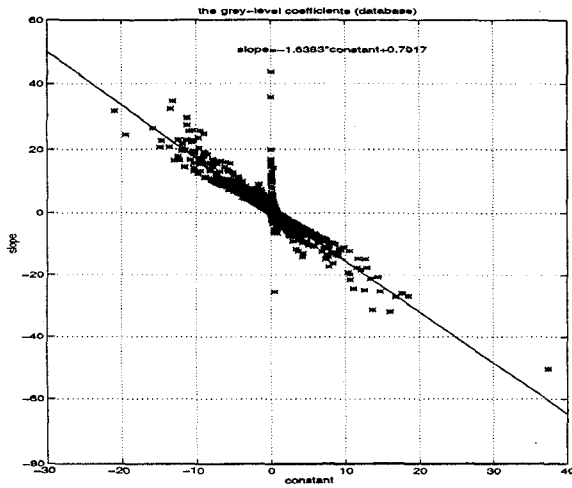


Figure 2: α vs. β coefficients for scheme I, based on four test images (“Lena”, “Ape”, “Aspen” and “San Francisco”).

quantization algorithm to quantize the slope and the intercept simultaneously as a two-dimensional vector. The LBG vector quantization algorithm [11] was used to quantize some of the highly correlated fractal coefficients simultaneously and dependently, and the results are illustrated in figures 3 and 4. The quantization details as well as the errors are given in the captions.

4. DISCUSSION AND CONCLUSION

This paper represents a preliminary attempt to study and exploit the redundancies in the fractal coefficients in order to achieve compression gain. For various fractal schemes, it was found that the grey-level coefficients exhibit some degree of linear dependence. The LBG vector quantizer was used to quantize the correlated coefficients simultaneously. Some of the benefits of using such a quantizer include (i) compression gain, (ii) preservation of edge information, (iii) fractal image compression at a predetermined bit rate, and (iv) lack of dependence on the image. The Huffman algorithm was used for binary coding of the quantized fractal coefficients, achieving a near-entropy bit rate performance.

In the fractal schemes I-III, a reduction in computational complexity of fractal transforms is generally accompanied by visible degradation in image fidelity. For the BFT scheme, however, a significant reduction in complexity was achieved with a relatively small loss in fidelity. Exploiting the correlation that seems to be inherent among some of the fractal coefficients results in an improvement of the compression capacity of these schemes. Further compression may be achieved by using differential and higher dimension LBG quantization



(a)



(b)

Figure 3: 2-Dimensional LBG quantization and Huffman coding of the fractal coefficients for “Lena” (512×512 pixels, 8 bits/pixel): (a) Scheme I: $L=4096$ levels (10 bits), $RMSE = 8.80$, $PSNR = 29.24$, Bit Rate = 0.359 bpp, $C_R=22.26:1$, encoding time = 3527 sec. (b) Scheme II: $L=4096$ levels (11 bits), $RMSE = 8.80$, $PSNR = 29.24$, Bit rate = 0.33 , $C_R = 24.38$, encoding time = 535 sec.

or decorrelation and orthogonal projection. Much work remains to be done in this regard.

5. ACKNOWLEDGEMENTS

This research was supported by an Interdisciplinary Research Grant from the University of Waterloo (ERV and AKK) as well as grants from the Natural Sciences and Engineering Council of Canada (NSERC) in the form of an individual Operating Grant (ERV) and a Collaborative Projects Grant (ERV). All support is gratefully acknowledged.

6. REFERENCES

- [1] M.F. Barnsley, *Fractals Everywhere*, New York: Academic Press, 1988.
- [2] M.F. Barnsley and S. Demko, "Iterated function systems and the global construction of fractals," *Proc. Roy. Soc. Lond.* A399, 243-275, 1985.
- [3] M.F. Barnsley and L.P. Hurd, *Fractal Image Compression*, Wellesley, MA: A.K. Peters, 1993.
- [4] Y. Fisher, editor, *Fractal Image Compression, Theory and Application*, New York: Springer Verlag, 1995.
- [5] Y. Fisher, Ed., *Fractal Image Encoding and Analysis*, Springer Verlag, Heidelberg, 1998.
- [6] A. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Proc.*, vol. 1, no. 1, pp. 18-30, 1992.
- [7] N. Lu, *Fractal Imaging*, Academic Press, New York, 1997.
- [8] S.G. Mallat and W.L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 617-643, 1992.
- [9] D. M. Monro, "A hybrid fractal transform," *Proc. ICASSP*, vo. 5, pp. 169-172, 1993.
- [10] D.M. Monro and S.J. Woolley, "Fractal image compression without searching", *Proc. ICASSP* 1994.
- [11] K. Sayood, *Introduction to Data Compression*, Morgan Kaufman, San Francisco, CA, 1996.
- [12] E.R. Vrscay, "A generalized class of fractal-wavelet transforms for image representation and compression," *Can. J. Elect. Comp. Eng.* vol. 23 (1998, to appear).



(a)



(b)

Figure 4: 2-Dimensional LBG quantization and Huffman coding of the fractal coefficients: (a) Scheme III: $L = 4096$ levels (11 bits), $RMSE = 12.35$, $PSNR = 26.29$, Bit rate = 0.17, $C_R = 46.55$, encoding time = 8 sec. (b) First order BFT, fractal coefficients $(\alpha, \beta, \gamma_x, \gamma_y)$, 1-D LBG Quantization of α , and γ_x independently with: $L = 512$ levels (9 bits), and 2-Dimensional LBG quantization of (β, γ_y) , simultaneously with: $L = 16384$ levels (12 bits), $RMSE = 10.11$, $PSNR = 28.03$, Bit rate = 0.47 bpp, $C_R = 17.01:1$, encoding time = 10 sec.